APPLICATION FOR
UNITED STATES PATENT
IN THE NAME OF

Steve Guttman
Joe Ternasky

for

CLIENT SIDE, WEB-BASED CALCULATOR

DOCKET NO. 22233.05481

DOCKET NO. 22233.05481

Please direct communications to:
Fenwick & West LLP
Two Palo Alto Square
Palo Alto, CA 94306
(650) 494-0600

Express Mail NumberEL566199665US

CLIENT SIDE, WEB-BASED CALCULATOR

Inventors: Steve Guttman and Joe Ternasky

This application claims priority under 35 U.S.C. §119(e) to U.S. Provisional Serial Number 60/241,083, of Guttman and Ternasky, filed October 16, 2000, which is herein incorporated by reference.

A INSB17 09/714,024

This application is related to U.S. Application Serial Number _____, of Guttman and Ternasky, filed November 15, 2000, entitled "Client Side, Web-Based Spreadsheet," which is herein incorporated by reference.

DOCKET: 22233-05481

Background of the Invention

The present invention relates generally to dynamic hypertext markup language (HTML) web pages and, more specifically, to a method for creating and viewing a calculator web page that has selectively viewable components.

Currently, web pages are defined using Hyper-Text Markup Language (HTML). Most web pages are passive. Specifically, a user just reads web pages and occasionally fills in a field of a form, submits the form to a server, and waits for a reply. In general, most users are not sophisticated consumers. Therefore, users want web-based programs to be fool-proof and easy to use. That is, web users want web-based programs to be designed so that the user cannot somehow damage the program or data by entering incorrect data or by clicking on a wrong area of the displayed web page.

Summary of the Invention

Active applications written in Dynamic HTML combine the look of a traditional desktop application with the facilities of the World Wide Web (WWW). When the user is working with a spreadsheet written in Dynamic HTML, the user can create formulas, add and delete columns, and format cells. At the same time, the user has a disadvantage of mistakenly changing content of any particular cell because each cell in a spreadsheet is editable. Thus, there is a need for a method and system that will allow working with a web-based spreadsheet so that the user is unable to change the content of any cell by mistake.

A described embodiment of the present invention allows a designer/creator to create and view "calculator" web pages. A calculator web page is a special version of a spreadsheet web page. When a user is working with a spreadsheet, it is unfortunately easy for him to mistakenly change the content of a particular cell because each cell in a spreadsheet is editable. The described embodiment of the present invention allows the user to view a spreadsheet as a calculator so that the user can work with a fill-in-the-blanks, standalone application. The user can just enter his data and view the calculator results. The user cannot change the content of any cell by mistakenly clicking on that cell. By default, the calculator uses a described method to determine which spreadsheet cells are initially editable in the calculator.

Another advantage of the described embodiment of the present invention is that in order for the user to work with a calculator, no other special software is involved, except for a web browser. The present invention serves as a design tool for the calculator designer because it produces interactive HTML pages without requiring the designer to

be familiar with the web programming. In accordance with the purpose of the present invention, as embodied and broadly described herein, the invention is a computer-implemented method for viewing a web-based spreadsheet in a calculator mode. The method comprises sending a request, by a user, to view the calculator web page; displaying the requested calculator web page, wherein only certain cells in the calculator are editable; and allowing the user to change contents in editable cells, but not allowing the user to change contents in non-editable cells.

In further accordance with the purpose of the invention, as embodied and broadly described herein, the invention is a computer-implemented method performed by a server data processing system. The method comprises receiving a request, from a client, for a calculator web page; reviewing parameters received with the request for the calculator web page, said parameters include a calculator mode and a data ID; sending the requested calculator web page, to the client, based on the parameters received, wherein said calculator web page contains embedded data specific to the requested calculator web page and wherein the calculator web page displays a calculator having only certain cells that are editable.

In further accordance with the purpose of the invention, as embodied and broadly described herein, the invention is a data structure comprising a member file for keeping information about members, a spreadsheet /calculator data file for keeping spreadsheet and calculator information on a server, and a Data Array file for keeping spreadsheet and calculator information on a client.

In further accordance with the purpose of the invention, as embodied and broadly described herein, the invention is a method for building a calculator, the method

comprising determining a number of rows and columns for the calculator; initializing the content of each cell in the calculator; making the cells visible once all the cells are built; and performing calculations for cells having formulas in them.

Advantages of the invention will be set forth in part in the description which follows and in part will be apparent from the description or may be learned by practice of the invention. The objects and advantages of the invention will be realized and attained by means of the elements and combinations particularly pointed out in the appended claims and equivalents.

BRIEF DESCRIPTION OF THE DRAWINGS

The accompanying drawings, which are incorporated in and constitute a part of this specification, illustrate several embodiments of the invention and, together with the description, serve to explain the principles of the invention.

Fig. 1 pictorially illustrates a web-based spreadsheet in a spreadsheet mode in accordance with a described embodiment of the present invention.

Fig. 2 illustrates a web-based calculator created using calculator preview mode in accordance with a described embodiment of the present invention.

Fig. 3 pictorially illustrates adjusting editability of cells in the calculator of Fig. 2 using calculator preview mode in accordance with a described embodiment of the present invention.

Fig. 4 pictorially illustrates the spreadsheet of Fig. 1 during use in a calculator mode in accordance with a described embodiment of the present invention.

09714683.111500

Fig. 5 is a block diagram illustrating an overall architecture in accordance with a described embodiment of the present invention.

Fig. 6A is a flow chart showing the process of returning an HTML web page to the user, performed by a server, in accordance with a described embodiment of the present invention.

Fig. 6B is a flow chart showing the steps performed by a browser in accordance with the present invention.

Fig. 7 is a flow chart illustrating the process of building a calculator in accordance with a described embodiment of the present invention.

Fig. 8 is a block diagram showing an example record in a member file in accordance with a described embodiment of the present invention.

Fig. 9 is a block diagram showing an example record in a spreadsheet /calculator data file in accordance with a described embodiment of the present invention.

Fig. 10 pictorially illustrates a process of allowing a user to embed a calculator in any web page in accordance with a described embodiment of the present invention.

Fig. 11 is a flow chart showing a log in process in accordance with a described embodiment of the present invention.

Fig. 12 is a flow chart showing a process of saving data to a server in accordance with a described embodiment of the present invention.

DOCKET ALARM

Fig. 13 pictorially illustrates a "My Files" page containing saved files in accordance with a described embodiment of the present invention.

Detailed Description of Illustrated Embodiments

Reference will now be made in detail to several embodiments of the present invention, examples of which are illustrated in the accompanying drawings. Wherever practicable, the same reference numbers will be used throughout the drawings to refer to the same or like parts.

A described embodiment of the present invention allows a user to view a web-based spreadsheet as a calculator, wherein only certain cells of the calculator are editable. The designer/creator of the web-based spreadsheet allows the user to alter the default as to which cells are editable when the calculator is displayed in a calculator preview mode. The described embodiment further allows the user to change the cells' content in the calculator mode based on the adjustments made in the calculator preview.

Fig. 1 illustrates a web-based spreadsheet in a spreadsheet mode. The spreadsheet 100 contains columns and rows where all the spreadsheet calculations are performed. Each row has a number and each column has a letter. A cell is the intersection of a row and a column and is referenced with a Column letter Row number notation, such as A1 or C3. The example spreadsheet 100 has at least 20 rows (numbered 1 through 20) and 5 columns (labeled A through E). The user can reset the number of rows and columns in the spreadsheet with the "Set Size" command. In addition, the user can add or delete columns with the "Insert/Delete Rows or Columns" commands. A cell can contain labels, numbers, text (strings), dates/times, or formulas. Other embodiments may have

additional types of data in their cells. A string is a text entry, such as, "Make Thanksgiving dinner," as displayed in cell A1 (104). A number is an integer or decimal value, which can be formatted as currency, integers (whole numbers), decimal numbers, or percentages. Users can define arbitrary formats.

The user can change information for any specific cell in the spreadsheet. To do so, the user has to click on that cell to highlight it. The command line 102 is the area of the spreadsheet where a user enters and edits cell values and formulas. To change information for a specific cell, a user clicks on that cell to highlight it. Then, the user can type the value, text or formula for that cell into the command line field. In the illustrated spreadsheet 100 the user entered the number "12" into the command line for cell B3 (106) and "2" for cell B5 (108). Also, the user chose a "traditional" dinner in cell 110 over "economy" or "alternate" dinner. If the user wants to change the type of meal he is planning (for example, the user wants to type "Corn Beef" instead of "Roast Turkey"), the user just needs to click on that cell to highlight it and enter "Corn Beef" into the command line. In the example spreadsheet, cell 111 has the following associated formula, as displayed in the command line 102: **=IF(B7="Traditional",1,25*(B3-B5),IF(B7="Alternate",(B3-B5),IF(B7="Economy",(B3-B5)*2,"")))**. Thus, if cell B7 contains "Traditional," a value of 12.5 is displayed in cell B13 (111), (as shown). This value is derived from the formula **1.25*(B3-B5),** wherein **B3=12** and **B5=2).** Different values of cell B7 cause other values to be displayed in cell B13 (111). Cells 112 also have formulas associated with them. Working with a spreadsheet allows a user to customize the spreadsheet for whatever purpose the user desires. When a user is

working with a spreadsheet, however, it is easy for him to mistakenly change content or formulas of a particular cell because each cell in a spreadsheet is editable.

Fig. 2 illustrates a web-based calculator created using calculator preview mode in accordance with a described embodiment of the present invention. Any spreadsheet can be viewed in either spreadsheet mode, calculator mode, or calculator preview mode. The drop-down "Calculator Preview" command, which is accessed from the spreadsheet mode, lets a designer/creator adjust which cells may be "filled-in" (edited) when the spreadsheet is viewed as a calculator. Viewing the spreadsheet in a spreadsheet mode, as illustrated in Fig. 1, allows a user to customize the spreadsheet for any purpose the user wishes. To pre-view the spreadsheet as a calculator, the user has to select the "Calculator Preview" command from the Tools menu. As shown in Fig. 2, the Menu Bar, Format Bar and Command Line will disappear. In place of the Format Bar an "Exit Preview" button (202) will appear. The spreadsheet grid is also hidden, and all cells that may be edited by a user of the calculator (106, 108, and 110) have a blue outline (or similar indication) around them.

By default, the system will take its best guess at which cells should be editable when the spreadsheet is displayed in a calculator preview mode. Cells that depend on values in other cells (for example, a cell with an associated formula (111)) default to non-editable. Cells with values that other cells depend on (106, 108, and 110) (but which don't have formulas in them) default to editable. Cells that do not depend on values in other cells default to non-editable.

Fig. 3 illustrates the process of allowing a designer/creator to adjust editability of cells in the spreadsheet of Fig. 1 using the calculator preview mode in accordance with a described embodiment of the present invention. As was illustrated in Fig. 2, by default, the system will take its best guess at which cells should be editable when the spreadsheet is initially displayed in calculator preview mode. Each Data Array file 502 and Spreadsheet/Calculator Data file 516 (as will be discussed with reference to Fig. 5) includes editability flags for each cell, which is assigned a "FALSE" or "TRUE" logic value. This value indicates whether a certain cell is editable or not. If the cell has a "TRUE" value, it is locked, and a user cannot edit that cell in the displayed calculator, in the calculator mode. Alternatively, if the cell has a "FALSE" value, it is unlocked, and a user of the displayed calculator can edit that cell. Thus, for example, in the displayed calculator of Fig. 4, only cells 106, 108, and 110 are unlocked and can be changed by a user in the calculator mode.

In some cases, however, a designer of a spreadsheet/calculator may want to adjust the default editability. To do so, the calculator designer/editor needs to click on a cell in calculator preview mode to toggle it from editable to non-editable and vice versa. Once the designer/creator clicks on the cell to adjust its editability, the matching object is found in the HTML page, and the executing JavaScript changes the editability flag value from "TRUE" to "FALSE" (or vice versa). For example, as shown in Fig. 2, a user viewing the calculator could only edit the cells related to the number of guests 106, the number of vegetarians 108, and the type of meal 110. Adjusting the cells' editability allows the user to edit the descriptive text or any other cell in the calculator when the calculator is displayed to the user. Once the designer/creator adjusts editability of cells, a user can

change the cells' values in a calculator mode. In particular, if the designer/creator toggles cell 304, which includes the text "Roast Turkey," in calculator preview mode the designer/creator changes its editability in the calculator preview mode so that a blue outline (or similar indication) appears around cell 304. Once in a calculator mode, the user can change the value of cell 304 by typing "Corn Beef" instead of "Roast Turkey". Similarly, toggling cell 302 in the calculator preview mode by the designer/creator allows the user to change the content of cell 302 in the calculator mode from "How many vegetarians" to "How many lacto-vegetarians." Thus, once the designer/creator has finished these modifications to editability, the user can change values of cells in the calculator mode based on the editability adjustments made in the calculator preview.

Fig. 4 illustrates an example of displaying the spreadsheet of Fig. 1 in a calculator mode in accordance with a described embodiment of the present invention. The example calculator web page 400 features a calculator displayed in calculator mode and in which only certain cells are editable. The calculator 400 allows the user to plan the Ultimate Holiday Dinner based on the number of guests and type of meal the user would like to make. The name of the calculator is displayed on the top of the page in the form of a text entry: "Make Thanksgiving dinner 1" 402. The user is prompted to enter information related to the number of total guests and the number of vegetarians among the guests 106 and 108. The user also is prompted to choose among traditional, alternate, or economy dinner 110. A number of total guests, number of vegetarians and a type of the dinner are the only editable cells in the calculator. These cells were set to be editable either by default when the spreadsheet was made or by the designer/creator in calculator preview mode.

In the bottom half of the page, the calculator displays the assortment of food that the user needs to have in order to prepare the dinner. It includes the following text entries, which are not editable: Roast Turkey, Bread Stuffing, Soup, Side of Fruits and Vegetables, Side of Potatoes, Crab Cake, and Cranberry Juice 406. In the calculator 400, the user is having twelve guests (where two people are vegetarians). The user has chosen a traditional dinner (110). Once the user enters all the requested information, the calculator estimates the quantity of food that the user needs to have in his kitchen in order to accommodate the total number of invited guests. It should be noted that the cells displaying the quantity of food 408 are not editable by the user. The calculator was so-designed by its designer/creator. The calculator determines that for the traditional type of dinner the user needs to have 12.5 pounds of roast turkey, 12 cups of bread stuffing, 12 cups of soup, 12 cups of vegetables on a side, 6 pounds of potatoes on a side, 2 pumpkin pies, and 24 cups of hot apple cider. Each of these cells has an associated formula depending on one or more of cells 106, 108, 110. The example calculator web page also includes a function menu 410 located at the top of the page allowing the user to save the calculator, print the calculator, receive instructions, edit calculator as a spreadsheet, email the calculator to others, and rate the calculator. In order to email the calculator, the user should click on "email this page" button and enter a recipient's address. The system will email a URL of the calculator, including the proper parameters in the URL.

Thus, the present invention advantageously allows working with a spreadsheet that is displayed in a calculator mode so that the user cannot change the content of any cell by mistakenly clicking on that cell. The calculator designer/creator can change the system default as to which cells are editable.

Fig. 5 is a block diagram illustrating an overall architecture of the present invention. A server system 510 includes, but is not limited to, a Member file 512, server software 514, a Spreadsheet /Calculator data file 516, a spreadsheet HTML 518, and a calculator HTML 521. It should be understood that the architecture illustrated in Fig. 5 is shown for purposes of example only and is not to be construed in a limiting sense.

The Member file 512 includes information for all members who have an account with the system. Initially, when a user logs into the system, he is asked whether he is a member who has an account with the system (the "log in" process will be discussed in detail with reference to Fig. 11). If the user is a member, he is allowed to access the system. In the alternative, if the user is not a member, he is offered to enter his identifying information, which might include a password and email address.

The Spreadsheet /Calculator data file 516 contains information related to spreadsheets and calculators. Such information may include a data ID, which refers to a file name, and information for each individual cell of a spreadsheet/calculator. In particular, the cell information related to each individual cell includes, but is not limited to, the information related to cell dependency, formatting, content, and editability. The server software 514 manages the files and communicates with the browser. The server software 514 handles opening, saving, and incorporating live data (such as stock quotes) into the files. In addition, the server software 514 returns the HTML page 518 and calculator page 521 in response to a request received from the client system 520, via a browser 504.

The client system 520 includes browser software 504 and a Data Array 502. The user, via the browser 504, sends a request to the server 510 for an HTML page. The

server software 514 reviews the parameters received with the request. If the parameters include a "spreadsheet" mode, a spreadsheet HTML page 518 is returned to the client 520. In the alternative, if the parameters include a "calculator" mode, the server 510 returns the calculator HTML page 520. If the parameters include an "embed" mode, the server returns a fully formed calculator web page. The Data Array 502 is an array of cell descriptions whose values are loadable from JavaScript in the HTML page. The Data Array 502 contains information related to each individual cell, which includes, but is not limited, to cell dependency, formatting, content, and editability. When the designer/creator saves a file, client 520 turns the definition of each cell into a string. The concatenated cell strings define the entire spreadsheet. Client 520 sends the concatenated string to the server 510. The server 510 takes the string and writes it into the Spreadsheet/Calculator Data File 516. Thus, a client-format Data Array 502 is translated into Spreadsheet/Calculator Data File 516 in the server file format.

The browser 504 is software effecting the requesting and displaying of HTML web pages. The browser software can be standalone or integrated within other software products. It should be understood that each of clients and web servers in the described embodiment preferably includes a processor and a memory. The memory includes instructions capable of being executed by the processor to perform the functions described below. The server can also include a computer readable medium for storing the instructions. The server system 510 communicates with the client system 520 via any appropriate communication mechanism, including but not limited to, a network, an intranet, the Internet, wireless communications, telecommunications, cable modems, and satellite communications.

Fig. 6A is a flow chart showing the process of returning an HTML web page, performed by the server 510, in accordance with a described embodiment of the present invention. In element 602, the server 510 receives a request for an HTML page, from the browser 504. In element 604, the server reviews the parameters received with the request (for example, the URL parameters in Common Gateway Interface (CGI) style). The parameters may include a data ID, which refers to the file name, and a mode, which preferably is one of spreadsheet mode, calculator mode, or embed mode. The server 510 finds the data in element 606 based on the data ID parameter received. If the designer/creator requested a spreadsheet, in element 608, the server embeds data using JavaScript and returns a spreadsheet HTML web page 518 to the designer/creator in element 610. If the designer/creator requested a calculator in element 612, the server 510 returns a calculator HTML web page 521 to the designer/creator in element 614. The spreadsheet HTML 518 contains script tags, spreadsheet data, and an HTML user interface. The calculator HTML 521 contains script tags, calculator data, and an HTML user interface. For example, in the case of a calculator HTML page 521, the script tag states the following, as shown in Table 1 (Table 1 shows an example of JavaScript included within a calculator web page, and is herein incorporated by reference): **href="/css/calculator_ie4_eda7fc0b.cssx"**, whereas a spreadsheet HTML page 518 script tag states: **href="/css/spreadsheet_ie4_630ddc74.cssx".** Furthermore, in the calculator HTML page 521, the user interface differs from that in a spreadsheet HTML 518. There are also some differences in JavaScript code embedded in calculator HTML page 521 and spreadsheet HTML 518. In particular, cells in a calculator HTML page 521 have different parameters than cells in a spreadsheet HTML page 518. Thus, cells in a

00571463.111501

spreadsheet are unlocked and the editability flag has a "FALSE" value, whereas most cells in a calculator are locked and the editability flag has a "TRUE" value.

If the designer/creator requested data in the Embed mode, a loaded, fully formed HTML page is returned, in element 616. This HTML page does not include HTML script tags, unlike calculator HTML 521 and spreadsheet HTML 518. The embedded HTML page dynamically returns the HTML page. This is accomplished using "JavaScript include" tags. The "JavaScript include" tags return dynamically created JavaScript reflecting the current change in the calculator or spreadsheet file. The process ends in 618 once the HTML page is returned to the designer/creator.

Fig. 6B is a flow chart showing the steps performed by a browser 520 in accordance with the present invention. Once the browser sends a request for an HTML page, as was discussed with reference to Fig. 6A, the server reviews the parameters received with the request and returns the HTML page accordingly. The parameters received may include a Data ID, which refers to the file name, and a mode, which preferably is one of spreadsheet mode, calculator mode, or embed mode. The browser 504, in turn, receives the HTML page, parses the page written in JavaScript, and executes JavaScript code, in element 626. The HTML page may include JavaScript files specific to spreadsheets and calculators, such as "Calculator JS file" and "Spreadsheet JS file" and shared data files, which could be shared by calculators and spreadsheets. Once the browser parses the page in element 626, it might send additional requests to the server 510 for data, which was referred to in the HTML page. In element 628, loading progress bar reports are generated and displayed informing the user about the status of the progress of program execution. In particular, the progress bar may indicate the following:

"Loading a Calculator" or "Loading a Spreadsheet." In element 630, the calculator is built (as shown in Fig. 7). In element 632, the user has the option of saving the file he is currently working with if the data was modified since the last time the file was saved. The process ends in element 634. Building the calculator 630 is discussed below.

Fig. 7 is a flow chart illustrating the process of building a calculator in accordance with a described embodiment of the present invention. Initially, a client system 520 receives the number of rows and columns to build a particular calculator, in element 702. For instance, as illustrated in reference with Fig. 4, an example calculator has 6 columns and 24 rows. A two-dimensional array of cell elements is built. In element 704, each cell is initialized and a dynamic HTML is specified for each cell. This step includes obtaining information about a particular cell from a Data Array file 502. This information includes, but is not limited to, cell dependency, formatting, content, and editability. Thus, as illustrated in Fig. 4, cell A3 which includes the text "HOW MANY TOTAL GUESTS ARE YOU HAVING?" is described by the following parameters, as shown in Table 1: **{ENTRY:'HOW MANY TOTAL GUESTS ARE YOU HAVING?',LOCKED:'FALSE',VIEWSIZE:'9PT',FORECOLOR:'NAVY',VIEWFAMILY:'VERDANA',_WRAPTEXT:'TRUE',_TEXTALIGN:'LEFT',M_ROW:3,M_COL:1,I_NR:'HOW MANY TOTAL GUESTS ARE YOU HAVING?'}**. According to these parameters, cell A3 is the intersection of a third row and a first column and is referenced with a Column # Row # notation A3. Cell A3 is an editable cell because the editability flag is set to "FALSE" (the calculator was so-designed by its designer/creator), the text in the cell is aligned to the left, the color of the cell is navy, and the size of the cell is 9pt.

INSB37

Once dynamic HTML is specified for each cell, the dependency tree is built, in step 705. The dependency tree is part of the Data Array 502. Any calculator has at least two types of cells: the ones that depend on a particular cell and the ones that a particular cell depends on. Some cells in the Data Array 502 contain the field "i_rt" and others contain the field "i_tb". The fields are lists of other cells that a given cell depends on (i_rt) and a list of cells that depend on this cell (i_tb). "I_tb" stands for "initialize the referred to by cell" and "i_rt" stands for "initialize refers to cell." In the example calculator illustrated in Fig. 3, cell B13 (111) is a dependent cell. The following is the list of other cells that cell B13 depends on, as shown in Table 1:

I_RT:'[E[7][2],E[3][2],E[5][2]]',LOCKED:'FALSE',VIEWSIZE:'9PT',_WIDTHCLUE:'80',_TEXTALIGN:'RIGHT',M_ROW:13,M_COL:2,I_NR:'12.5'}. Accordingly, cell B13, which is the intersection of row 13 and column 2, depends on cells B7 (110), B3 (106), and B5 (108), where cell B7 displays the type of dinner the user selected, cell B3 displays the number of guests the user is expecting, and cell B5 displays the number of vegetarians among the guests. Thus, if cell B7 contains "Traditional," a value of 12.5 is displayed in cell B13 (111), (as shown in Fig. 3). This value is derived from the formula, which is described in the Data Array 502: **1.25*(B3-B5).** If B3=12 and B5=2, the value displayed in cell B13 is 12.5. If cell B7 contains "Economy," a value of 20 is displayed in cell B13 (not shown). This value is derived from the formula described in the Data Array 502 **(B3-B5)*2.** If B3=12 and B5=2, the value displayed in cell B13 is 20.

Once all cells are built, they become visible, in element 706. This is done with a combination of HTML that is generated on the fly using JavaScript and by JavaScript itself. In element 708, calculations are performed for each cell, when needed. For

example, if the user chooses a "traditional" dinner in cell B7, cell A13 displays a "Roast Turkey." Formula calculations are performed for cell B13 to determine how many pounds of roast turkey are required to prepare the dinner based on the number of guests. If the user chooses an "alternate" dinner in cell B7, cell A13 displays "Cornish Game Hens." Calculations are performed for cell B13 to determine how many pieces of hen are required to prepare the dinner. Similarly, if the user chooses "economy" dinner in cell B7, cell A13 displays "Chicken Parmesan." Calculations are performed for cell B13 to determine how many pieces of chicken are required to accommodate twelve guests, where two of them are vegetarians.

Once all calculations are made, the determination is made whether a particular cell is editable (whether it is locked), in element 710. If a cell is non-editable, that cell is locked and the user is not allowed to enter data into that cell in calculator mode, in element 712. For example, according to Data Array 502 description, cells A13-A23 are locked and the user cannot change their content. In the alternative, if the cell is editable (unlocked) (for example, cells B3 and B5, as described in the Data Array 502), an input box is built around that cell, in element 714. The user can type in the value in that cell because it is unlocked in calculator mode, in element 716. Thus, the user can type in "12" in cell B3 and "2" in cell B5. The process ends in 718.

Fig. 8 is a block diagram showing a Member file 512 and an example member record 800. The Member file 512 keeps member information in the form of member records 800. The member record 800 identifies a particular member. In one embodiment of the present invention, the example member record may have the following fields: a member ID, a password, an email address, and a data ID, which refers

to a file name. The example record 800 features a member ID equal to ANNA, 0124 as a password, ANNA@HOTMAIL.COM, as an email address, and "Make Thanksgiving dinner 1" as a file name. When a user initially signs up with the system, all user information is forwarded to the member file 512.

Fig. 9 is a block diagram showing a Spreadsheet /Calculator data file 516 and an example record 900. The Spreadsheet /Calculator data file 516 keeps spreadsheet and calculator data information. The same data file is used to generate spreadsheet HTML 518 and calculator HTML 521. In particular, it contains a user name, data ID (for example, a file name) and information about each cell. The information for every cell may include value, formula, formatting, editability, and borders. Each Data Array file 502 and Spreadsheet/Calculator Data file 516 includes an editability flag for each cell, which is assigned a "FALSE" or "TRUE" logic value. This value indicates whether a certain cell is editable or not. If the cell has a "TRUE" value, i.e., if it is locked, a user cannot edit that cell when viewing the calculator. Alternatively, if the cell has a "FALSE" value, i.e., if it is unlocked, that cell can be edited when viewing the calculator. Fig. 9 features the example record 900 having the following fields: User Name = ANNA, Data ID= "Make Thanksgiving Dinner," and information about each cell. In particular, cell A3 which includes the text "HOW MANY TOTAL GUESTS ARE YOU HAVING?" is described by the following parameters, as shown in Table 1: **{ENTRY:'HOW MANY TOTAL GUESTS ARE YOU HAVING?',LOCKED:'FALSE',VIEWSIZE:'9PT',FORECOLOR:'NAVY',VIEWFAMILY:'VERDANA',_WRAPTEXT:'TRUE',_TEXTALIGN:'LEFT',M_ROW:3,M_COL:1,I_NR:'HOW MANY TOTAL GUESTS ARE YOU HAVING?'}**. According to these parameters, cell A3 is the

intersection of a third row and a first column and is referenced with a Column letter Row number notation A3. Cell A3 is an editable cell because the editability flag is set to "FALSE" (the calculator was so-designed by its designer/creator), the text in the cell is aligned to the left, the color of the cell is navy, and the size of the cell is 9pt.

Fig. 10 illustrates a process of allowing a web page user to embed a calculator in any web page. If a user has a personal web site, he can include one of his calculators with other text and graphics directly into his web site. To embed a calculator in a web page, the user has to go to the "My Files" page (see Fig. 13). Then, the user has to highlight the file he wants to embed and select the "Sharing" command from the Action Panel on the right side of the screen. Performing this step will bring up a dialog box with the appropriate HTML code (which includes the parameters of data ID for the calculator and anchor text that will appear as a link on the page). The user has to copy this code (1002) to the clipboard and then paste the code into his web page's HTML at the location he wants the calculator to appear. In the Sharing dialog, the user first needs to set his file to Public. Embedding the calculator advantageously allows to view the calculator file within the user's own web page. When a web page designer/creator embeds a calculator on a web page, he is actually adding some code to an HTML page which calls that calculator from the server 510. When this code is downloaded to someone's web browser, the browser makes a request to the server 510 to display the calculator of interest. The server then returns the appropriate calculator and Dynamic HTML code to that web browser. As discussed above in reference to Fig. 6A, the HTML page does not include HTML script tags, unlike calculator HTML 521 and spreadsheet HTML 518. The embedded HTML page includes means to dynamically return the HTML content. These

means are "JavaScript include tags. " The JavaScript include tags return dynamically created JavaScript reflecting the current change in the calculator or spreadsheet file.

The embedded calculator is a fully formed calculator web page which looks identical to the calculator web page discussed in reference to Fig. 4. The example calculator web page 400 features a calculator, wherein only certain cells are editable. The calculator 400 allows the user to plan Thanksgiving dinner based on the number of guests and type of meal the user would like to make. The name of the calculator is displayed right on the top of the page in the form of a text entry: "Make Thanksgiving dinner" (402). The user is prompted to enter information related to the number of total guests and the number of vegetarians among the guests 106 and 108. Also, the user is prompted to choose among traditional, alternate, or economy dinner 110. A number of total guests, number of vegetarians and a type of the dinner are the only editable cells in the calculator. These cells were set to be editable either by default when the spreadsheet was made or by the designer/creator in calculator preview mode.

Fig. 11 is a flow chart showing a log in process. The process starts in element 1100 and determines whether the user is a private user, in element 1102. The user has the option of making any individual file a "public" file, so that he can share it with others (see Fig. 10). A private user can only view private files if he has logged into the account where the file is listed. Alternatively, a public user has the URL of the file and wants to be able to view the file outside the private user's personal account. If the user is a private user, the process determines whether the user is a member, in element 1104. At this point, the user's name and password are posted to the server. The server 510 reviews the parameters and verifies that the user has an account with the system, i.e., the user is a

member. If the user is not a member, the ERROR message is returned in element 1108, and the user is prompted to sign up with the system, in element 1118. In the alternative, if the user is a member, as determined from member file 512, the system determines, in element 1106, whether the user is the owner of the file he would like to work with. Every Spreadsheet/Calculator data file 516 contains a member ID. If the user ID matches the member ID in the Spreadsheet/Calculator data file 516, the user is the file owner. If the user is a file owner the system allows the user to view that file, in element 1112, and the process ends in 1114. Similarly, if the user is a public user a public user who has the URL of the file and wants to be able to view the file outside the private user's personal account, the system allows the public user to view that file, in element 1112.

If the member is not a file owner, the ERROR message is returned in element 1110 and the user is prompted to work with that particular file as not a file owner, in element 1116. If the user is a public user (he has the URL of the file and wants to be able to view the file outside the private user's personal account) he can view that file, in element 1112.

Fig. 12 is a flow chart showing a process of saving data into "My Files." The *Save* and *Save As* commands, which are accessed from the FILE menu, allow a calculator designer/creator to save the file he is currently working with. *Save* will overwrite the last saved version of the file, where *Save As* renames the current file and saves it (under a new name). The process starts in element 1202. In element 1204, the process determines whether calculator designer/creator changes editability. If the designer/creator changes editability, the editability flag is changed from "FALSE" to "TRUE" in client-side Data Array 502, in element 1206. If designer/creator changes cell's content in a spreadsheet,

in element 1208, data content is changed in the Data Array 502, in element 1210. If the creator/designer indicates that he wants to save new data, in element 1212, the server 510 receives a "post" in the form of new data in client-side format, from a browser, in element 1214. In element 1216, the server determines whether the creator/designer is a member who has an account with the system in accordance with member file 512. If the user is a member, the process determines, in element 1218, whether the user is the owner of the file he is working with. In the alternative, if the user is not a member, in element 1222 the user is prompted to register and become a member. If the member is the owner of the file he is working with, the process offers the user to save the information in place of the original file, in element 1220. In the alternative, if the user is not the owner of the file, he is offered to save the file as not a file owner, in element 1224. The process then saves information about each cell, including all attributes, such as format, borders, editability, and cell dependency as well as rows height and columns width into non-owner account in element 1226. The process ends in element 1228.

Fig. 13 illustrates "My Files" page containing saved files. The "My Files" page 1300 is a user's personal file manager where each member accesses and manages his files. To access the files from the "My Files" page 1300, the user has to enter his name and password. The user has the option of making any individual file a "public" file, so that he can share it with others. The upper left panel 1302 of the page lets the user navigate between different major areas of the site. The middle area 1304 of the page displays user's list of files. The menu bar 1306 directly above the list of files allows the user to create new files, import files and images, sort the list of files, and view the files according to their type.

While the invention has been described in conjunction with a specific embodiment, it is evident that many alternatives, modifications and variations will be apparent to those skilled in the art in light of the foregoing description. Accordingly, the present invention is intended to embrace all such alternatives, modifications and variations as fall within the spirit and scope of the appended claims and equivalents.

TABLE 1

```
<html>
 <head>
      <TITLE>make thanksgiving dinner1 - Blox.com BrainMatter
Calculator</TITLE>

      <link rel="stylesheet" type="text/css"
href="/css/static/view_mode_ie_67c1e830.cssx"/><link rel="stylesheet"
type="text/css" href="/css/calculator_ie4_eda7fc0b.cssx"/>

            <style type="text/css">
            /* /home/www/template/static/view_mode_ie.style.tpl */
      #buttonDivision {top: 30; left: 0;}
      #buttonPalette {top: 0; left: 0;height: 30;visibility: visible;}
      #topDivision {top: 0; left: 0;visibility: hidden;}

            </style>


            <script language="JavaScript">

                  var kTotalJSIncludes = 13;
                  var gNumJSIncludesLoaded = 0;
                  var kFromGallery = false;
                  var kDomain = "blox.com";
                  var kRootUrl = "http://www.blox.com/";

                  // hide 'rate this calc' button if we're not opening from
the gallery
                  if (!kFromGallery)
```

Docket No. 22233-05481

```
suppressElements(['btnRate','btnRateText','break5']);

                        function JSIncludeDoneLoading()
                        {
                                gNumJSIncludesLoaded++;
                                includeModalProgressWidget.style.width =
(gNumJSIncludesLoaded/kTotalJSIncludes)*250;
                        }

                        // creates stylesheet rules to suppress the specified
elements
                        function suppressElements(elemIdArray) {
                                if (!(document.styleSheets &&
document.styleSheets.length > 0)) return;
                                for (var i=0; i < elemIdArray.length; i++)
                                        document.styleSheets[0].addRule('#' +
elemIdArray[i], 'display:none');
                        }


                </script>

        </head>

        <body class="body" scroll="no" onLoad="loadDocument()"
onBeforeUnload="if(typeof unloadDocument != 'undefined')unloadDocument()"
onUnload="if(typeof unloadDocument != 'undefined')unloadDocument()"
onResize="resizeDocument()">

                <span class='view' id='view' style="width: 100%; height: 100%;">
```

```
<!-- later write the contents of this span -->
</span>

<script language="JavaScript">

        var content = '<div class="modalLabel">Please wait a moment
(the first time you use BrainMatter this may take a little longer).</div>'
        + '<div class="modalProgressBar">'
        + '<div id="includeModalProgressWidget"
class="modalProgressWidget" style="visibility: visible;">'
        + '</div></div>';

        var progressDialog = '<table id="includeProgressDialog"
width="100%" height="100%">'
        + '<tr valign="middle"><td align="center"><div
class="modal" STYLE="visibility:visible; position:static;">'
        + '<table border="0" width="225" cellspacing="0"
cellpadding="0">'
        + '<tr valign="middle">'
        + '<td bgcolor="#ffcc00"><div
class="modalTopLeftBox"></div></td>'
        + '<td bgcolor="#ffe57d" nowrap><span
class="modalTitle">Loading BrainMatter</span></td>'
        + '</tr>'
        + '<tr>'
        + '<td bgcolor="#ffe57d"><br></td>'
        + '<td><div class="modalContent">' + content + '</div><div
id="modalBottomBar" style="height:10px; overflow:hidden"><BR></div></td>'
        + '</tr>'
        + '</table>'
        + '</div></tr></td></table>';
```

```
                document.write(progressDialog);

            </script>

            <script language="JavaScript">

                // global mode flag (can be "design", "view", or "embed")
                var kMode = "view";
                var gInitDirtyCells = "";
                var gCellElemMatrix; // 2-D array of the cell elements, use
like: gCellElemMatrix[row][column]
                var gRowCount, gColumnCount, gImported=0 , gOwnerId=0,
gAppId='',gAppDescription='';
                var gAppVersion = 1.0;
                var gAppAuthor = 'ss';
                var gMacroText = '';
                var gPartnerMenu = null;
                var gPopupMenu = null;
                var gExtra = "";
                var gIsMatterhorn = false;
                var gSaveScript = "savebraincell";
                var gSaveLifeboat = "lifeboat";

                //
        gColumnCount = 6;gRowCount = 24;


                // handlers should be placed on outermost control...
                function GetColumnWidthsAndHeights(widths, heights)
                {
```

DOCKET ALARM

```
                //
widths[1] = 399;
widths[2] = 84;
widths[3] = 73;
heights[2] = 19;
heights[3] = 19;
heights[4] = 19;
heights[5] = 19;
heights[6] = 19;
heights[7] = 31;
heights[8] = 19;
heights[9] = 19;
heights[10] = 19;
heights[11] = 19;
heights[12] = 19;
heights[13] = 19;
heights[14] = 19;
heights[15] = 19;
heights[16] = 19;
heights[17] = 19;
heights[18] = 19;
heights[19] = 19;
heights[20] = 31;
heights[21] = 19;
heights[22] = 19;
heights[23] = 19;

        }

        // after this comment, we add //init cells
gAppId='003d8cf3_9815';
```

09741683.111500

```
gOwnerId='003d76ac_435d';
gAppDescription='make thanksgiving dinner1';
var gCellDataArray=[{entry:'Make Thanksgiving dinner',viewSize:'12pt',foreColor:'navy',viewFamily:'Verdana',_fontWeight:'bold',_wrapText:'true',_textAlign:'left',m_row:1,m_col:1,i_nr:'Make Thanksgiving dinner'},
{viewSize:'9pt',foreColor:'navy',viewFamily:'Verdana',_wrapText:'true',_textAlign:'left',m_row:2,m_col:1},
{entry:'How many total guests are you having?',viewSize:'9pt',foreColor:'navy',viewFamily:'Verdana',_wrapText:'true',_textAlign:'left',m_row:3,m_col:1,i_nr:'How many total guests are you having?'},
{entry:'12',dynamic:'12',derived:'12',
i_tb:'[e[13][2],e[14][2],e[19][2],e[15][2],e[17][2],e[21][2],e[23][2]]',viewSize:'9pt',_widthClue:'80',foreColor:'maroon',_textAlign:'right',m_row:3,m_col:2,i_nr:'12'},
{viewSize:'9pt',foreColor:'navy',viewFamily:'Verdana',_wrapText:'true',_textAlign:'left',m_row:4,m_col:1},
{foreColor:'maroon',viewSize:'9pt',_textAlign:'right',m_row:4,m_col:2},
{entry:'How many vegetarians?',viewSize:'9pt',foreColor:'navy',viewFamily:'Verdana',_wrapText:'true',_textAlign:'left',m_row:5,m_col:1,i_nr:'How many vegetarians?'},
{entry:'2',dynamic:'2',derived:'2',
i_tb:'[e[13][2]]',viewSize:'9pt',_widthClue:'80',foreColor:'maroon',_textAlign:'right',m_row:5,m_col:2,i_nr:'2'},
{viewSize:'9pt',foreColor:'navy',viewFamily:'Verdana',_wrapText:'true',_textAlign:'left',m_row:6,m_col:1},
{foreColor:'maroon',viewSize:'9pt',_textAlign:'right',m_row:6,m_col:2},
{entry:'Traditional Holiday Dinner, or something a little more unusual? (Traditional, Alternate or Economy)',viewSize:'9pt',foreColor:'navy',viewFamily:'Verdana',_wrapText:'true',
```

_textAlign:'left',m_row:7,m_col:1,i_nr:'Traditional Holiday Dinner, or something a little more unusual? (Traditional, Alternate or Economy)'},

{entry:'=menu("Traditional","traditional","Alternate","Alternate","Economy","economy")',dynamic:'__menu("r7c2s0", \'Traditional\',\'traditional\',\'Alternate\',\'Alternate\',\'Economy\',\'economy\')',derived:'traditional',

i_tb:'[e[14][1],e[17][1],e[19][1],e[21][1],e[23][1],e[13][2],e[13][3],e[17][3],e[17][2],e[11][1],e[20][1],e[13][1]]',m_cellGUI:'{type:\'menu\', setting:0, label:\'Traditional\'}',locked:'true',viewSize:'9pt',m_dirtyOnInit:'true',foreColor:'maroon',_textAlign:'left',m_row:7,m_col:2,i_nr:'Traditional'},

{viewSize:'9pt',foreColor:'navy',viewFamily:'Verdana',_wrapText:'true',_textAlign:'left',m_row:8,m_col:1},

{locked:'true',m_row:8,m_col:2},

{entry:'Here\'s what you can whip up in your kitchen:',viewSize:'9pt',foreColor:'navy',viewFamily:'Verdana',_wrapText:'true',_textAlign:'left',m_row:9,m_col:1,i_nr:'Here\'s what you can whip up in your kitchen:'},

{viewSize:'9pt',foreColor:'navy',viewFamily:'Verdana',_wrapText:'true',_textAlign:'left',m_row:10,m_col:1},

{entry:'=IF(B7 ="Economy","A Basic Holiday Feast",if(B7 ="Alternate","An "& B7 &" Holiday Feast","A "& B7 &" Holiday Feast"))',dynamic:'(_eqr(_cel(\'r7c2s0\'),\'Economy\')?\'A Basic Holiday Feast\':(_eqr(_cel(\'r7c2s0\'),\'Alternate\')?_concat(_concat(\'An \',_cel(\'r7c2s0\')),\' Holiday Feast\'):_concat(_concat(\'A \',_cel(\'r7c2s0\')),\' Holiday Feast\')))',derived:'A traditional Holiday Feast',

i_rt:'[e[7][2]]',viewSize:'9pt',foreColor:'navy',viewFamily:'Verdana',_fontWeight:'bold',_wrapText:'true',_textAlign:'left',m_row:11,m_col:1,i_nr:'A traditional Holiday Feast'},

{_borderBottom:'solid black 1pt',viewSize:'9pt',foreColor:'navy',viewFamily:'Verdana',_wrapText:'true',_textAlign:'left',m_row:12,m_col:1},

{_borderBottom:'solid black 1pt',viewSize:'9pt',_textAlign:'right',m_row:12,m_col:2},
{textAlign:'left',_borderBottom:'solid black 1pt',viewSize:'9pt',m_row:12,m_col:3},
{entry:'=IF(B7 ="Traditional","Roast Turkey",IF(B7 ="Alternate","Cornish Game Hens",IF(B7 ="Economy","Chicken Parmesan","")))',dynamic:'(_eqr(_cel(\'r7c2s0\'),\'Traditional\')?\'Roast Turkey\':(_eqr(_cel(\'r7c2s0\'),\'Alternate\')?\'Cornish Game Hens\':(_eqr(_cel(\'r7c2s0\'),\'Economy\')?\'Chicken Parmesan\':\'\')))',derived:'Roast Turkey',
i_tb:'[e[15][1]]',
i_rt:'[e[7][2]]',viewSize:'9pt',foreColor:'navy',viewFamily:'Verdana',_fontWeight:'bold',_wrapText:'true',_textAlign:'left',m_row:13,m_col:1,i_nr:'Roast Turkey'},
{entry:'=IF(B7 ="Traditional",1.25*(B3 - B5),IF(B7 ="Alternate",(B3 - B5),IF(B7 ="Economy",(B3 - B5)*2,"")))',dynamic:'(_eqr(_cel(\'r7c2s0\'),\'Traditional\')?_mul(1.25,_sub(_cel(\'r3c2s0\'),_cel(\'r5c2s0\'))):(_eqr(_cel(\'r7c2s0\'),\'Alternate\')?_sub(_cel(\'r3c2s0\'),_cel(\'r5c2s0\')):(_eqr(_cel(\'r7c2s0\'),\'Economy\')?_mul(_sub(_cel(\'r3c2s0\'),_cel(\'r5c2s0\')),2):\'\')))',derived:'12.5',
i_rt:'[e[7][2],e[3][2],e[5][2]]',viewSize:'9pt',_widthClue:'80',_textAlign:'right',m_row:13,m_col:2,i_nr:'12.5'},
{entry:'=IF(B7 ="Traditional","pounds",IF(B7 ="Alternate","hens",IF(B7 ="Economy","pieces","")))',dynamic:'(_eqr(_cel(\'r7c2s0\'),\'Traditional\')?\'pounds\':(_eqr(_cel(\'r7c2s0\'),\'Alternate\')?\'hens\':(_eqr(_cel(\'r7c2s0\'),\'Economy\')?\'pieces\':\'\')))',derived:'pounds',

DOCKET ALARM

i_rt:'[e[7][2]]',textAlign:'left',viewSize:'9pt',m_row:13,m_col:3,i_nr:'pounds'
},
{entry:'=IF(B7 ="Traditional","Bread Stuffing",IF(B7
="Alternate","Corn Bread",IF(B7 ="Economy","Garlic
Breadsticks",""))))',dynamic:'(_eqr(_cel(\'r7c2s0\'),\'Traditional\')?\'Bread
Stuffing\':(_eqr(_cel(\'r7c2s0\'),\'Alternate\')?\'Corn Bread\':(_eqr(_cel(
\'r7c2s0\'),\'Economy\')?\'Garlic Breadsticks\':\'\')))',derived:'Bread
Stuffing',
i_tb:'[e[14][2],e[14][3]]',
i_rt:'[e[7][2]]',viewSize:'9pt',foreColor:'navy',viewFamily:'Verdana',_fontW
eight:'bold',_wrapText:'true',_textAlign:'left',m_row:14,m_col:1,i_nr:'Bread
Stuffing'},
{entry:'=IF(A14 ="Garlic Breadsticks", B3 *2,if(A14
="","", B3))',dynamic:'(_eqr(_cel(\'r14c1s0\'),\'Garlic
Breadsticks\')?_mul(_cel(\'r3c2s0\'),2):(_eqr(_cel(\'r14c1s0\'
),\'\')?\'\':_cel(\'r3c2s0\')))',derived:'12',
i_rt:'[e[14][1],e[3][2]]',viewFormat:'number',viewSize:'9pt',_widthClue:'80',
_textAlign:'right',m_row:14,m_col:2,i_nr:'12'},
{entry:'=IF(A14 ="Bread Stuffing","cups",if(A14
="","","pieces"))',dynamic:'(_eqr(_cel(\'r14c1s0\'),\'Bread
Stuffing\')?\'cups\':(_eqr(_cel(\'r14c1s0\'
),\'\')?\'\':\'pieces\'))',derived:'cups',
i_rt:'[e[14][1]]',textAlign:'left',viewSize:'9pt',m_row:14,m_col:3,i_nr:'cups'},
{entry:'=if(A13 ="","","Soup")',dynamic:'(_eqr(_cel(\'r13c1s0\'
),\'\')?\'\':\'Soup\')',derived:'Soup',
i_tb:'[e[16][1],e[15][2],e[15][3]]',
i_rt:'[e[13][1]]',viewSize:'9pt',foreColor:'navy',viewFamily:'Verdana',_font
Weight:'bold',_wrapText:'true',_textAlign:'left',m_row:15,m_col:1,i_nr:'Soup'},
{entry:'=if(A15 ="","", B3)',dynamic:'(_eqr(_cel(
\'r15c1s0\'),\'\')?\'\':_cel(\'r3c2s0\'))',derived:'12',

DOCKET ALARM

i_rt:'[e[15][1],e[3][2]]',viewSize:'9pt',_widthClue:'80',_textAlign:'right',m_row:15,m_col:2,i_nr:'12'},

{entry:'=if(A15 ="","","cups")',dynamic:'(_eqr(_cel(\'r15c1s0\'),\'\')?\'\':\'cups\')',derived:'cups',

i_rt:'[e[15][1]]',textAlign:'left',viewSize:'9pt',m_row:15,m_col:3,i_nr:'cups'},

{entry:'=if(A15 ="","," - try a simple corn chowder or spicy bean soup")',dynamic:'(_eqr(_cel(\'r15c1s0\'),\'\')?\'\':\' - try a simple corn chowder or spicy bean soup\')',derived:' - try a simple corn chowder or spicy bean soup',

i_rt:'[e[15][1]]',viewSize:'9pt',foreColor:'navy',viewFamily:'Verdana',_wrapText:'true',_textAlign:'left',m_row:16,m_col:1,i_nr:'- try a simple corn chowder or spicy bean soup'},

{entry:'=IF(B7 ="Traditional","Side of Vegetables",IF(B7 ="Alternate","Stuffed Mushrooms",IF(B7 ="Economy","Side of Vegetables","")))',dynamic:'(_eqr(_cel(\'r7c2s0\'),\'Traditional\')?\'Side of Vegetables\':(_eqr(_cel(\'r7c2s0\'),\'Alternate\')?\'Stuffed Mushrooms\':(_eqr(_cel(\'r7c2s0\'),\'Economy\')?\'Side of Vegetables\':\'\')))',derived:'Side of Vegetables',

i_tb:'[e[17][3],e[18][1],e[17][2]]',

i_rt:'[e[7][2]]',viewSize:'9pt',foreColor:'navy',viewFamily:'Verdana',_fontWeight:'bold',_wrapText:'true',_textAlign:'left',m_row:17,m_col:1,i_nr:'Side of Vegetables'},

{entry:'=IF(B7 ="Alternate",3* B3 ,if(A17 ="","",1* B3))',dynamic:'(_eqr(_cel(\'r7c2s0\'),\'Alternate\')?_mul(3,_cel(\'r3c2s0\')):(_eqr(_cel(\'r17c1s0\'),\'\')?\'\':_mul(1,_cel(\'r3c2s0\'))))',derived:'12',

i_rt:'[e[7][2],e[3][2],e[17][1]]',viewFormat:'number',viewSize:'9pt',_widthClue:'80',_textAlign:'right',m_row:17,m_col:2,i_nr:'12'},

{entry:'=IF(B7 ="Alternate","mushrooms",if(A17 ="","","cups"))',dynamic:'(_eqr(_cel(\'r7c2s0\'

),\'Alternate\')?\'mushrooms\':(_eqr(_cel(\'r17c1s0\'
),\'\')?\'\':\'cups\'))',derived:'cups',
i_rt:'[e[7][2],e[17][1]]',textAlign:'left',viewSize:'9pt',m_row:17,m_col:3,i_nr:'cups'},
{entry:'=if(A17 ="","," - try green beans, squash, or a vegetable medley as well")',dynamic:'(_eqr(_cel(\'r17c1s0\'),\'\')?\'\':\' - try green beans, squash, or a vegetable medley as well\')',derived:' - try green beans, squash, or a vegetable medley as well',
i_rt:'[e[17][1]]',viewSize:'9pt',foreColor:'navy',viewFamily:'Verdana',_wrapText:'true',_textAlign:'left',m_row:18,m_col:1,i_nr:'- try green beans, squash, or a vegetable medley as well'},
{entry:'=IF(B7 ="Traditional","Side of Potatoes",IF(B7 ="Alternate","Yams",IF(B7 ="Economy","Yams",""))))',dynamic:'(_eqr(_cel(\'r7c2s0\'),\'Traditional\')?\'Side of Potatoes\':(_eqr(_cel(\'r7c2s0\'),\'Alternate\')?\'Yams\':(_eqr(_cel(\'r7c2s0\'),\'Economy\')?\'Yams\':\'\')))',derived:'Side of Potatoes',
i_tb:'[e[19][2],e[19][3],e[20][1]]',
i_rt:'[e[7][2]]',viewSize:'9pt',foreColor:'navy',viewFamily:'Verdana',_fontWeight:'bold',_wrapText:'true',_textAlign:'left',m_row:19,m_col:1,i_nr:'Side of Potatoes'},
{entry:'=if(A19 ="","", B3 /2)',dynamic:'(_eqr(_cel(\'r19c1s0\'),\'\')?\'\':_div(_cel(\'r3c2s0\'),2))',derived:'6',
i_rt:'[e[19][1],e[3][2]]',viewFormat:'number',viewSize:'9pt',_widthClue:'80',_textAlign:'right',m_row:19,m_col:2,i_nr:'6'},
{entry:'=if(A19 ="","","pounds")',dynamic:'(_eqr(_cel(\'r19c1s0\'),\'\')?\'\':\'pounds\')',derived:'pounds',
i_rt:'[e[19][1]]',textAlign:'left',viewSize:'9pt',m_row:19,m_col:3,i_nr:'pounds'},
{entry:'=IF(B7 ="Traditional"," - there\'s more than one way to cook a potato! Try scalloped, mashed, twice-baked, or even au gratin!",if(A19

="",""," - yams are sweeter than potatoes and can be prepared just as easily, for a different twist on the holidays!",))',dynamic:'(_eqr(_cel(\'r7c2s0\'
),\'Traditional\')?\' - there\\\'s more than one way to cook a potato! Try scalloped, mashed, twice-baked, or even au gratin!\':(_eqr(_cel(\'r19c1s0\'
),\'\')?\'\':\' - yams are sweeter than potatoes and can be prepared just as easily, for a different twist on the holidays!\'))',derived:' - there\'s more than one way to cook a potato! Try scalloped, mashed, twice-baked, or even au gratin!',
i_rt:'[e[7][2],e[19][1]]',viewSize:'9pt',foreColor:'navy',viewFamily:'Verdana',_wrapText:'true',_textAlign:'left',m_row:20,m_col:1,i_nr:'- there\'s more than one way to cook a potato! Try scalloped, mashed, twice-baked, or even au gratin!'},
{entry:'=IF(B7 ="Traditional","Pumpkin Pie",IF(B7 ="Alternate","Pecan Pie",IF(B7 ="Economy","Pumpkin Pie","")))',dynamic:'(_eqr(_cel(\'r7c2s0\'),\'Traditional\')?\'Pumpkin Pie\':(_eqr(_cel(\'r7c2s0\'),\'Alternate\')?\'Pecan Pie\':(_eqr(_cel(\'r7c2s0\'),\'Economy\')?\'Pumpkin Pie\':\'\')))',derived:'Pumpkin Pie',
i_tb:'[e[21][2],e[21][3],e[22][1]]',
i_rt:'[e[7][2]]',viewSize:'9pt',foreColor:'navy',viewFamily:'Verdana',_fontWeight:'bold',_wrapText:'true',_textAlign:'left',m_row:21,m_col:1,i_nr:'Pumpkin Pie'},
{entry:'=if(A21 ="","",ROUND(B3 /8,0))',dynamic:'(_eqr(_cel(\'r21c1s0\'),\'\')?\'\':__round(_div(_cel(\'r3c2s0\'
),8),0))',derived:'2',
i_rt:'[e[21][1],e[3][2]]',viewFormat:'number',viewSize:'9pt',_widthClue:'80',_textAlign:'right',m_row:21,m_col:2,i_nr:'2'},
{entry:'=if(A21 ="","","pies")',dynamic:'(_eqr(_cel(\'r21c1s0\'
),\'\')?\'\':\'pies\')',derived:'pies',
i_rt:'[e[21][1]]',textAlign:'left',viewSize:'9pt',m_row:21,m_col:3,i_nr:'pies'},
{entry:'=if(A21 ="",""," - or try something different, like a Sweet Potato Pie!")',dynamic:'(_eqr(_cel(\'r21c1s0\'),\'\')?\'\':\' - or try something different, like a Sweet Potato Pie!\')',derived:' - or try something different, like a Sweet Potato Pie!',

```
i_rt:'[e[21][1]]',viewSize:'9pt',foreColor:'navy',viewFamily:'Verdana',_wrap
Text:'true',_textAlign:'left',m_row:22,m_col:1,i_nr:'- or try something different,
like a Sweet Potato Pie!'},
        {entry:'=IF(   $B$7   ="Traditional","Hot Apple Cider",IF(    $B$7
        ="Alternate","Spiced Cranberry Cider",IF(      $B$7   ="Economy","Hot
Apple Cider","")))',dynamic:'(_eqr(_cel( \'r7c2s0\'      ),\'Traditional\')?\'Hot
Apple Cider\':(_eqr(_cel(    \'r7c2s0\'      ),\'Alternate\')?\'Spiced Cranberry
Cider\':(_eqr(_cel(    \'r7c2s0\'      ),\'Economy\')?\'Hot Apple
Cider\':\'\')))',derived:'Hot Apple Cider',
        i_tb:'[e[23][2],e[23][3]]',
        i_rt:'[e[7][2]]',_borderBottom:'solid black
1px',viewSize:'9pt',foreColor:'navy',viewFamily:'Verdana',_fontWeight:'bold',_wra
pText:'true',_textAlign:'left',m_row:23,m_col:1,i_nr:'Hot Apple Cider'},
        {entry:'=if(    A23    ="","",       B3      *2)',dynamic:'(_eqr(_cel(
        \'r23c1s0\'     ),\'\')?\'\':_mul(_cel( \'r3c2s0\'      ),2))',derived:'24',
        i_rt:'[e[23][1],e[3][2]]',_borderBottom:'solid black
1px',viewFormat:'number',viewSize:'9pt',_widthClue:'80',_textAlign:'right',m_row
:23,m_col:2,i_nr:'24'},
        {entry:'=if(    A23    ="","","cups")',dynamic:'(_eqr(_cel(      \'r23c1s0\'
        ),\'\')?\'\':\'cups\')',derived:'cups',
        i_rt:'[e[23][1]]',textAlign:'left',_borderBottom:'solid black
1px',viewSize:'9pt',m_row:23,m_col:3,i_nr:'cups'}];

        var needsRebuilding=false;
        var gDerivedFrom='';
        var kSaveHostname = "";
        gMacroText= "";
```

```
</script>

<script language="JavaScript" src="/lib/js/dialog_ie_d6af9e21.jsx"
></script><script language="JavaScript"
src="/lib/js/browser_detect_ie_6af40f80.jsx" ></script><script
language="JavaScript" src="/lib/js/cookie_ie_81818cb9.jsx" ></script><script
language="JavaScript" src="/lib/js/gui_ie_fc55692d.jsx" ></script><script
language="JavaScript" src="/lib/js/document_ie_ec836dad.jsx" ></script><script
language="JavaScript" src="/lib/js/calculator_ie_b5a1d7ed.jsx" ></script><script
language="JavaScript" src="/lib/js/interface_calculator_ie_f84b9570.jsx"
></script><script language="JavaScript" src="/lib/js/runtime_ie_38489e48.jsx"
></script><script language="JavaScript" src="/lib/js/cell_ie_f1e68dc0.jsx"
></script><script language="JavaScript" src="/lib/js/compiler_ie_574f1484.jsx"
></script><script language="JavaScript" src="/lib/js/platform_ie_bb68a125.jsx"
></script><script language="JavaScript" src="/lib/js/resource_ie_b70be76d.jsx"
></script><script language="JavaScript" src="/lib/js/utility_ie_16a663aa.jsx"
></script>

<script language="JavaScript">

    function WindowIsValid(wnd) {
        try {
            var x = wnd.name;
        }
        catch (error) {
            return 0;
        }
        return 1;
    }
```

```
function SafeEval(str) {
        try {
                eval( str );
        }
        catch (error) {
                return error.description ?
error.description : "unknown error";
        }
        return 0;
}
</script>


<div id="topDivision" class="division"></div>
<div id="buttonPalette" class="palette">
<a href="http://www.blox.com/" target="_blank"
onFocus="this.blur()"><img src="/img/blox_cube_16x26.gif" width=16 height=26
vspace=2 hspace=8 border=0 alt="Blox.com"></a><div id="break0"
class="break"></div>
<div id="btnSave" class="button" title="Save"></div><div
id="btnSaveText" class="buttonText">Save</div>
<div id="break1" class="break"></div>
<div id="btnPrint" class="button" title="Print"></div><div
id="btnPrintText" class="buttonText">Print</div>
<div id="break2" class="break"></div>
<div id="btnInstruction" class="button" title="Instructions"></div><div
id="btnInstructionText" class="buttonText">Instructions</div>
<div id="break3" class="break"></div>
<div id="btnEdit" class="button" title="Edit as spreadsheet"></div><div
id="btnEditText" class="buttonText">Edit as spreadsheet</div>
<div id="break4" class="break"></div>
```

00511463.111502

```
        <div id="btnEmail" class="button" title="Email to others"></div><div
id="btnEmailText" class="buttonText">Email to others</div>
        <div id="break5" class="break"></div>
        <div id="btnRate" class="button" title="Rate this calculator"></div><div
id="btnRateText" class="buttonText">Rate this calculator</div>
        </div>
        <div id="buttonDivision" class="division"></div>
        <div id="footDivision" class="division"></div>
        <div id="footPalette" class="palette">
        <nobr>Copyright &copy; 1999-2000 <a href="http://www.alphablox.com"
target="_blank">AlphaBlox Corporation Inc.</a> &trade;</nobr></div>

                <div id="modalDialog" style="display:none"></div>
                <div id="popupMenuList" class="smallList" style="width:100;
height:1;"></div>

                <script language="JavaScript">
                        includeProgressDialog.style.display = "none";
                </script>

                <iframe id=resource0 src="" style="display: none;"></iframe>
                <iframe id=resource1 src="" style="display: none;"></iframe>
                <iframe id=resource2 src="" style="display: none;"></iframe>
                <iframe id=resource3 src="" style="display: none;"></iframe>

                <iframe id="saveFrame" src="/save_frame.html" style="display:
none;"></iframe>
                <iframe id="macroFrame" src="/macro_frame" style="display:
none;"></iframe>
                <iframe id="bugFrame" src="" style="display: none;"></iframe>
```

09754800.1

```
            <iframe name="postFrame" src="" style="display:
none;"></iframe>
              <form id="postForm" method="POST"></form>
          </body>
          </html>
```

09174883.1